FACULTATEA DE MATEMATICA SI INFORMATICA

UNIVERSITATEA DIN BUCURESTI

**PROIECT APLICATII PROFESIONALE IN BAZE DE DATE**

**Profesor indrumator Autori**

Lect.dr. Velcescu Letitia Cernea Teodora

Bogdan Mihai Timofte

Bucuresti, 2015

**CUPRINS**

**1. Descrierea modelului real**  3

**2. Constrangeri**  4

**3. Descrierea enitatilor, relatiilor, atriburelor** 4

3.1. Entitati 4

3.2. Relatii 6

3.3. Atribute 10

**4. Scheme relationale** 17

**5. Normalizare** 19

**6. Denormalizare**  20

**7. Crearea bazei de date** 21

**8. Ce se doreste de la model**  29

**9. Diagrama conceptuala a modelului** 29

Modelarea unei retele de socializare

1. Descrierea modelului real

Proiectul invoca crearea unui model responsabil cu reprezentarea informatiilor referitoare la gestiunea unei retele de socializare.

Principalul actor al modelului nostru relational este utilizatorul (USER) ce are posibilitatea de a realiza diferite tipuri de actiuni in sistem. In primul rand, acesta este caracterizat de un profil (PROFILE) pentru care poate configura adresa, orasul (CITY), judetul (COUNTY), precum si tara (COUNTRY). De asemenea, i se acorda dreptul de a-si alege limba (LANGUAGE).

Interactiunea unui utilizator cu alti utilizatori da nastere conceptului de lista de prieteni (FRIEND\_LIST). Aceasta schiteaza relatia de prietenie virtuala dintre doua sau mai multe persoane, ce se realizeaza prin intermediul cererilor de prietenie. Mai mult de atat, utilizatorii aplicatiei pot comunica prin intermediul mesajelor (MESSAGE).

Datele cu caracter personal (profil, adresa, localitate, judet, tara), precum si pozele sau albumele unui utilizator sunt protejate prin optiunea (PRIVACY), ramanand la decizia acestuia de a le face publice, private sau vizibile doar de catre prieteni.

Dintre actiunile pe care le poate intreprinde un utilizator in aplicatie, enumeram :

* Publicarea unei stari (STATUS);
* Crearea si publicarea unui album (ALBUM);
* Incarcarea si postarea unei poze (PICTURE);
* Adaugarea, editarea, stergerea unui comentariu (COMMENT) – adaugat la o poza (COMMENT\_PICTURE), la un album (COMMENT\_ALBUM) sau la un status (COMMENT\_STATUS);
* Aprecierea (LIKE) – a unei poze (LIKE\_PICTURE), a unui album (LIKE\_ALBUM), a unui status (LIKE\_STATUS) sau a unui comentariu (LIKE\_COMMENT) postate de alti utilizatori ce fac parte din lista de prieteni.
* Crearea si/sau participarea (PARTICIPATE\_EVENT) la un eveniment (EVENT).

Fiecare utilizator este la curent cu actiunile realizate de prietenii sai prin intermediul stirilor (FEED). In cadrul stirilor-ului apar inregistrari legate de pozele (FEED\_PICTURE), albumele (FEED\_ALBUM) create, precum si de modificarile de status-uri (FEED\_STATUS) existente. Mai mult, utilizatorii sunt instiintati si de evenimentele create (FEED\_EVENT) de catre alti prieteni.

Aplicatia pune la dispozitie si un modul de notificari (NOTIFICATION) prin care fiecare utilizator este instiintat de interactiunile directe ale altor utilizatori. Acestea se pot realiza prin intermediul comentariilor (NOTIFICATION\_COMMENT) sau like-urilor (NOTIFICATION\_LIKE) si se reflecta asupra pozelor, albumelor si asupra statusurilor utilizatorului. Acceptarea unei cereri de prietenie trimise e redata de o notificare de aprobare a prieteniei (NOTIFICATION\_FRIEND).

1. Constrangeri

* Un user nu se poate imprieteni cu el insusi
* Un user are un singur profil.
* Un utilizator poate crea unul sau mai multe albume.
* Un album contine una sau mai multe poze.
* Mesajele dintre utilizatorii care nu sunt prieteni vor fi marcate ca fiind SPAM.

1. Descrierea entitatilor, relatiilor si atributelor

In urmatoarea sectiune vor fi prezentate entitatile, relatiile si atributele ce duc la modelarea unei retele de socializare ce are functionalitatile mai sus amintite.

* 1. . Entitati

**USER** = Cheie primara : user\_id. Entitatea modeleaza si inregistreaza in aplicatie personajul principal : utilizatorul aplicatiei. Acesta este caracterizat de cateva date personale.

**PROFILE** = Cheie primara : profile\_id. Profilul unui utilizator contine informatii despre preferintele acestuia, educatia si obiceiurile sale.

**CITY** = Cheie primara : city\_id. Identifica orasul in care locuieste utilizatorul.

**COUNTY** = Cheie primara : county\_id. Judet sau sector in care locuieste utilizatorul.

**COUNTRY** = Cheie primara : country\_id. Tara in care locuieste utilizatorul.

**LANGUAGE** = Cheie primara : language\_id. Limba de circulatie pe care utilizatorul o seteaza.

**FRIEND\_LIST** = Cheie primara : friend\_list\_id. Lista ce retine relatiile de prietenie dintre doi sau mai multi utilizatori.

**MESSAGE** = Cheie primara : message\_id. Reprezinta mesajele ce sunt transmise intre utilizatorii ce sunt prieteni.

**STATUS** = Cheie primara : status\_id. Starea de spirit, gandurile pe care un utilizator le impartaseste virtual cu prietenii sai.

**PRIVACY** = Cheie primara : primary\_id. Optiune disponibila utilizatorului prin care acesta isi poate configura setarile privind publicul tinta al postarilor sale: public, privat, etc.

**ALBUM** = Cheie primara : album\_id. Albumul pe care il creeaza utilizatorul si care contine pozele incarcate de acesta.

**PICTURE** = Cheie primara : picture\_id. Poza pe care utilizatorul o poate incarca in mediul online impreuna cu o descriere.

**COMMENT** = Cheie primara : comment\_id. Mesaj pe care un utilizator il poate lasa - public la poza, albumul sau statusul altui utilizator.

**COMMENT\_PICTURE** = Cheie primara : comment\_picture\_id. Comentariul pe care un utilizator il poate lasa la poza incarcata de alt utilizator.

**COMMENT\_ALBUM** = Cheie primara : comment\_album\_id. Comentariul pe care un utilizator il poate lasa la albumul creat de alt utilizator.

**COMMENT\_STATUS** = Cheie primara : comment\_status\_id. Comentariul pe care un utilizator il poate lasa la statusul unui utilizator.

**LIKE** = Cheie primara : like\_id. Reprezinta semnul de apreciere al unui utilizator fata de postarile altui utilizator.

**LIKE\_PICTURE** = Cheie primara : like\_picture\_id. Aprecierea unei poze.

**LIKE\_ALBUM** = Cheie primara : like\_album\_id. Aprecierea unui album.

**LIKE\_STATUS** = Cheie primara : like\_status\_id. Aprecierea unui status.

**LIKE\_COMMENT** = Cheie primara : like\_comment\_id. Aprecierea unui status.

**EVENT** = Cheie primara : event\_id. Identifica evenimentele de natura sociala, educationala ce exista in sistem.

**PARTICIPATE\_EVENT** = Cheie primara : participate\_event\_id. Participarea la un eveniment.

**FEED** = Cheie primara : feed\_id. Noutatile postate de alti utilizatori pe care un utilizator le vede.

**FEED\_PICTURE** = Cheie primara : feed\_picture\_id. Pozele noi pe care le-au incarcat alti utilizatori si pe care le vede un utilizator.

**FEED\_ALBUM** = Cheie primara : feed\_album\_id. Albumele noi create de catre alti utilizatori pe care le vede un utilizator.

**FEED\_STATUS**  = Cheie primara : feed\_status\_id. Statusurile noi pe care le creeaza sau modifica alti utilizatori si pe care le vede un utilizator.

**FEED\_EVENT** = Cheie primara : feed\_event\_id. Evenimentele noi create si participarea altor utilizatori la acestea pe care un utilizator le vede.

**NOTIFICATION** = Cheie primara: notification\_id. Notificarile ce apar pe contul unui utilizator.

**NOTIFICATION\_COMMENT** – Cheie primara : notification\_comment\_id. Notificarea pe care o primeste un utilizator atunci cand prietenii sai ii lasa comentarii la postari (albume, poze, statusuri).

**NOTIFICATION\_LIKE** – Cheie primara : notification\_like\_id. Notificarea pe care o primeste un utilizator atunci cand prietenii sai apreciaza una dintre postarile sale (albume, poze, statusuri, comentarii).

**NOTIFICATION\_FRIEND** – Cheie primara : notification\_friend\_id. Notificarea pe care o primeste un utilizator atunci cand un alt utilizator ii trimite sau accepta o cerere de prietenie.

* 1. Relatii

**USER\_are\_PROFILE**: relatie ce leaga entitatile USER si PROFILE. Cardinalitate minima: 1:1. Cardinalitate maxima: 1:1.

**USER\_FRIEND\_LIST**: relatie ce leaga entitatile USER si FRIEND\_LIST. Cardinalitate minima: 1:0. Cardinalitate maxima: 1:n.

**USER\_trimite\_MESSAGE**: relatie ce leaga entitatile USER si MESSAGE. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**PROFILE\_are\_CITY**: relatie ce leaga entitatile PROFILE si CITY. Cardinalitatea minima: 0:0. Cardinalitatea maxima: n:1.

**CITY\_apartine\_COUNTY**: relatie ce leaga entitatile CITY si COUNTY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: n:1.

**COUNTY\_apartine\_COUNTRY**: relatie ce leaga entitatile COUNTY si COUNTRY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: n:1.

**USER\_seteaza\_LANGUAGE**: relatie ce leaga entitatile USER si LANGUAGE. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_posteaza\_STATUS**: relatie ce leaga entitatile USER si STATUS. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**USER\_creeaza\_ALBUM**: relatie ce leaga entitatile USER si ALBUM. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**ALBUM\_contine\_PICTURE**: relatie ce leaga entitatile USER si PICTURE. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**USER\_LIKE\_PICTURE**: relatie ce leaga entitatile USER si LIKE\_PICTURE. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**LIKE\_PICTURE \_PICTURE**: relatie ce leaga entitatile LIKE\_PICTURE si PICTURE. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_LIKE\_ALBUM**: relatie ce leaga entitatile USER si LIKE\_ALBUM. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**LIKE\_ALBUM \_ALBUM**: relatie ce leaga entitatile LIKE\_ALBUM si ALBUM. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_LIKE\_STATUS**: relatie ce leaga entitatile USER si LIKE\_STATUS. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**LIKE\_STATUS \_STATUS**: relatie ce leaga entitatile LIKE\_STATUS si STATUS. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_LIKE\_COMMENT**: relatie ce leaga entitatile USER si LIKE\_COMMENT. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**LIKE\_COMMENT \_COMMENT**: relatie ce leaga entitatile LIKE\_COMMENT si COMMENT. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**LIKE\_LIKE\_PICTURE**: relatie ce leaga entitatile LIKE si LIKE\_PICTURE. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**LIKE\_LIKE\_ALBUM**: relatie ce leaga entitatile LIKE si LIKE\_ALBUM. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**LIKE\_LIKE\_STATUS**: relatie ce leaga entitatile LIKE si LIKE\_STATUS. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**LIKE\_LIKE\_COMMENT**: relatie ce leaga entitatile LIKE si LIKE\_COMMENT. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**USER\_COMMENT\_PICTURE**: relatie ce leaga entitatile USER si COMMENT\_PICTURE. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**COMMENT\_PICTURE \_PICTURE**: relatie ce leaga entitatile COMMENT\_PICTURE si PICTURE. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_COMMENT\_ALBUM**: relatie ce leaga entitatile USER si COMMENT\_ALBUM. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**COMMENT\_ALBUM \_ALBUM**: relatie ce leaga entitatile COMMENT\_ALBUM si ALBUM. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_COMMENT\_STATUS**: relatie ce leaga entitatile USER si COMMENT\_STATUS. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**COMMENT\_STATUS \_STATUS**: relatie ce leaga entitatile COMMENT\_STATUS si STATUS. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**COMMENT\_COMMENT\_PICTURE**: relatie ce leaga entitatile COMMENT si COMMENT\_PICTURE. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**COMMENT\_COMMENT\_ALBUM**: relatie ce leaga entitatile COMMENT si COMMENT\_ALBUM. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**COMMENT\_COMMENT\_STATUS**: relatie ce leaga entitatile COMMENT si COMMENT\_STATUS. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**PICTURE\_are\_PRIVACY**: relatie ce leaga entitatile PICTURE si PRIVACY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: 1:1.

**ALBUM\_are\_PRIVACY**: relatie ce leaga entitatile ALBUM si PRIVACY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: 1:1.

**STATUS\_are\_PRIVACY**: relatie ce leaga entitatile STATUS si PRIVACY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: 1:1.

**PROFILE\_are\_PRIVACY**: relatie ce leaga entitatile PROFILE si PRIVACY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: 1:1.

**CITY\_are\_PRIVACY**: relatie ce leaga entitatile CITY si PRIVACY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: 1:1.

**COUNTY\_are\_PRIVACY**: relatie ce leaga entitatile COUNTY si PRIVACY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: 1:1.

**COUNTRY\_are\_PRIVACY**: relatie ce leaga entitatile COUNTRY si PRIVACY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: 1:1.

**COUNTRY\_are\_PRIVACY**: relatie ce leaga entitatile COUNTRY si PRIVACY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: 1:1.

**USER\_creeaza\_EVENT**: relatie ce leaga entitatile USER si EVENT. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**USER\_PARTICIPATE\_EVENT**: relatie ce leaga entitatile USER si PARTICIPATE\_EVENT. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**PARTICIPATE\_EVENT \_EVENT**: relatie ce leaga entitatile PARTICIPATE\_EVENTsi EVENT. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**EVENT\_are\_CITY**: relatie ce leaga entitatile EVENT si CITY. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_vizualizeaza\_FEED\_PICTURE**: relatie ce leaga entitatile USER si FEED\_PICTURE. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**FEED\_PICTURE\_contine\_PICTURE**: relatie ce leaga entitatile FEED\_PICTURE si PICTURE. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_vizualizeaza\_FEED\_ALBUM**: relatie ce leaga entitatile USER si FEED\_ALBUM. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**FEED\_ALBUM\_contine\_ALBUM**: relatie ce leaga entitatile FEED\_ALBUM si ALBUM. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_vizualizeaza\_FEED\_STATUS**: relatie ce leaga entitatile USER si FEED\_STATUS. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**FEED\_STATUS\_contine\_STATUS**: relatie ce leaga entitatile FEED\_STATUS si STATUS. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_vizualizeaza\_FEED\_EVENT**: relatie ce leaga entitatile USER si FEED\_EVENT. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**FEED\_EVENT\_contine\_EVENT**: relatie ce leaga entitatile FEED\_EVENT si EVENT. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**FEED\_FEED\_PICTURE**: relatie ce leaga entitatile FEED si FEED\_PICTURE. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**FEED\_FEED\_ALBUM**: relatie ce leaga entitatile FEED si FEED\_ALBUM. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**FEED\_FEED\_STATUS**: relatie ce leaga entitatile FEED si FEED\_STATUS. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**FEED\_FEED\_EVENT**: relatie ce leaga entitatile FEED si FEED\_EVENT. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**USER\_primeste\_NOTIFICATION\_LIKE**: relatie ce leaga entitatile USER si NOTIFICATION\_LIKE. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**NOTIFICATION\_LIKE\_contine\_LIKE**: relatie ce leaga entitatile NOTIFICATION\_LIKE si LIKE. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_primeste\_NOTIFICATION\_COMMENT**: relatie ce leaga entitatile USER si NOTIFICATION\_COMMENT. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**NOTIFICATION\_COMMENT\_contine\_COMMENT**: relatie ce leaga entitatile NOTIFICATION\_COMMENT si COMMEN T. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**NOTIFICATION\_FRIEND\_FRIEND\_LIST**: relatie ce leaga entitatile NOTIFICATION\_FRIEND si FRIEND\_LIST. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**NOTIFICATION\_NOTIFICATION\_LIKE:** relatie ce leaga entitatile NOTIFICATION si NOTIFICATION\_LIKE. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**NOTIFICATION\_NOTIFICATION\_COMMEN T:** relatie ce leaga entitatile NOTIFICATION si NOTIFICATION\_COMMENT. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**NOTIFICATION\_NOTIFICATION\_FRIEND:** relatie ce leaga entitatile NOTIFICATION si NOTIFICATION\_FRIEND. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

* 1. Atribute

**USER**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| user\_id | number(6) | Cheie primara |
| language\_id | number(6) | Cheie externa, facand legatura cu limba pe care utilizatorul o seteaza |
| username | nvarchar2(255) | Nume ales de utilizator |
| password | nvarchar2(255) | Parola aleasa de utilizator |
| first\_name | nvarchar2(255) | Prenumele utilizatorului |
| last\_name | nvarchar2(255) | Numele utilizatorului |
| email | nvarchar2(255) | E-mailul utilizatorului |
| active | nchar(1) | Starea contului: activ sau inactiv |
| created\_at | date | Data crearii contului utilizatorului |

**PROFILE**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| profile\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul. |
| privacy\_id | number(6) | Cheie externa, realizand legatura cu PRIVACY |
| city\_id | number(6) | Cheie externa, realizand legatura cu orasul |
| date\_of\_birth | date | Data de nastere a utilizatorului |
| about\_me | nvarchar2(255) | Descrierea utilizatorului |
| relationship | nvarchar2(255) | Starea civila |
| phone | nvarchar2(255) | Numarul de telefon |
| interests | nvarchar2(255) | Interesele utilizatorului |
| education | nvarchar2(255) | Educatia utilizatorului |
| hobbies | nvarchar2(255) | Pasiunile, obiceiurile utilizatorului |
| fav\_movies | nvarchar2(255) | Filmele preferate de utilizator |
| fav\_artists | nvarchar2(255) | Artistii preferati de utilizator |
| fav\_books | nvarchar2(255) | Cartile preferate de utilizator |
| address | nvarchar2(255) | Adresa utilizatorului |

**CITY**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| city\_id | number(6) | Cheie primara |
| county\_id | number(6) | Cheie externa, realizand legatura cu judetul |
| privacy\_id | number(6) | Cheie externa, realizand legatura cu PRIVACY |
| name | nvarchar2(255) | Numele orasului |

**COUNTY**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| county\_id | number(6) | Cheie primara |
| country\_id | number(6) | Cheie externa, realizand legatura cu tara |
| privacy\_id | number(6) | Cheie externa, realizand legatura cu PRIVACY |
| name | nvarchar2(255) | Numele judetului |

**COUNTRY**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| country\_id | number(6) | Cheie primara |
| privacy\_id | number(6) | Cheie externa, realizand legatura cu PRIVACY |
| name | nvarchar2(255) | Numele tarii |

**LANGUAGE**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| language\_id | number(6) | Cheie primara |
| name | nvarchar2(255) | Numele limbii de circulatie |

**FRIEND\_LIST**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| friend\_list\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul |
| friend\_user\_id | number(6) | Cheie externa, realizand legatura cu prietenul utilizatorului |
| state | nchar(1) | Starea relatiei de prietenie: acceptata sau nu |
| created\_at | date | Data trimiterii cererii de prietenie de catre un utilizator |

**MESSAGE**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| message\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care trimite/ primeste mesajul |
| friend\_user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care primeste/ trimite mesajul |
| message\_content | nvarchar2(255) | Continutul mesajului |
| created\_at | date | Data crearii mesajului |
| is\_read | nchar(1) | Starea mesajului: citit sau nu |
| is\_spam | nchar(1) | Tipul mesajului: spam sau nu |

**STATUS**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| status\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul |
| privacy\_id | number(6) | Cheie externa, realizand legatura cu PRIVACY |
| status\_content | nvarchar2(255) | Continutul statusului |
| created\_at | date | Data crearii statusului |
| updated\_at | date | Data editarii/ modificarii statusului |
| numberOfLikes | int | Numarul de like-uri detinute de un status |

**PRIVACY**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| privacy\_id | number(6) | Cheie primara |
| privacy\_type | nvarchar2(255) | Tip: public, vizualizat doar de mine (only me), vizualizat doar de prietenii mei (only friends) |

**PICTURE**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| picture\_id | number(6) | Cheie primara |
| album\_id | number(6) | Cheie externa, realizand legatura cu albumul din care face parte poza |
| privacy\_id | number(6) | Cheie externa, realizand legatura cu PRIVACY |
| description | nvarchar2(255) | Descrierea pozei |
| created\_at | date | Data incarcarii pozei |
| updated\_at | date | Data editarii, modificarii descrierii pozei |
| is\_album\_cover | nchar(1) | Daca este sau nu coperta albumului din care face parte |
| is\_profile\_picture | nchar(1) | Daca este sau nu poza de profil |
| numberOfLikes | int | Numarul de like-uri detinute de o poza |

**ALBUM**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| album\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul |
| privacy\_id | number(6) | Cheie externa, realizand legatura cu PRIVACY |
| name | nvarchar2(255) | Numele albumului |
| description | nvarchar2(255) | Descrierea albumului |
| created\_at | date | Data crearii albumului |
| updated\_at | date | Data editarii, modificarii descrierii sau numelui albumului |
| numberOfLikes | int | Numarul de like-uri detinute de un album |

**COMMENT**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| comment\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a postat comentariul |
| comment\_type\_id | number(6) | Cheie externa, realizand legatura cu tipul comentariului |
| comment\_type | nvarchar2(255) | Tipul comentariului |

**COMMENT\_PICTURE**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| comment\_picture\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a postat comentariul |
| picture\_id | number(6) | Cheie externa, realizand legatura cu poza la care a comentat utilizatorul |
| comment\_content | nvarchar2(255) | Continutul comentariului |
| created\_at | date | Data postarii comentariului |
| updated\_at | date | Data editarii, modificarii comentariului |
| numberOfLikes | int | Numarul de like-uri detinute de un comentariu |

**COMMENT\_ALBUM**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| comment\_album\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a postat comentariul |
| album\_id | number(6) | Cheie externa, realizand legatura cu albumul la care a comentat utilizatorul |
| comment\_content | nvarchar2(255) | Continutul comentariului |
| created\_at | date | Data postarii comentariului |
| updated\_at | date | Data editarii, modificarii comentariului |
| numberOfLikes | int | Numarul de like-uri detinute de un comentariu |

**COMMENT\_STATUS**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| comment\_status\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a postat comentariul |
| status\_id | number(6) | Cheie externa, realizand legatura cu statusul la care a comentat utilizatorul |
| comment\_content | nvarchar2(255) | Continutul comentariului |
| created\_at | date | Data postarii comentariului |
| updated\_at | date | Data editarii, modificarii comentariului |
| numberOfLikes | int | Numarul de like-uri detinute de un comentariu |

**LIKE**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| like\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a dat LIKE |
| like\_type\_id | number(6) | Cheie externa, realizand legatura cu tipul de like |
| like\_type | nvarchar(2) | Tipul de like |

**LIKE\_PICTURE**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| like\_picture\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a dat LIKE |
| picture\_id | number(6) | Cheie externa, realizand legatura cu poza pe care a apreciat-o utilizatorul |
| created\_at | date | Data la care un utilizator a dat LIKE |

**LIKE\_ALBUM**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| like\_album\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a dat LIKE |
| album\_id | number(6) | Cheie externa, realizand legatura cu albumul pe care l-a apreciat utilizatorul |
| created\_at | date | Data la care un utilizator a dat LIKE |

**LIKE\_STATUS**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| like\_status\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a dat LIKE |
| status\_id | number(6) | Cheie externa, realizand legatura cu statusul pe care l-a apreciat utilizatorul |
| created\_at | date | Data la care un utilizator a dat LIKE |

**LIKE\_COMMENT**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| like\_comment\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a dat LIKE |
| comment\_id | number(6) | Cheie externa, realizand legatura cu comentariul pe care l-a apreciat utilizatorul |
| created\_at | date | Data la care un utilizator a dat LIKE |

**EVENT**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| event\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a creat evenimentul |
| city\_id | number(6) | Cheie externa, realizand legatura cu orasul evenimentului |
| name | nvarchar2(255) | Numele evenimentului |
| description | nvarchar2(255) | Descrierea evenimentului |
| address | nvarchar2(255) | Adresa evenimentului |
| date | date | Data la care are loc evenimentul |
| created\_at | date | Data crearii evenimenului |
| updated\_at | date | Data modificarii evenimentului |

**PARTICIPATE\_EVENT**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| participate\_event\_id | number(6) | Cheie primara |
| event\_id | number(6) | Cheie externa, realizand legatura cu un eveniment |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care participa la un eveniment |

**FEED**

|  |  |  |
| --- | --- | --- |
| Nume | Tip | Descriere |
| feed\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul |
| feed\_type\_id | number(6) | Cheie externa, realizand legatura cu tipul FEED-ului |
| feed\_type | nvarchar2(255) | Tipul Feed-ului |

**FEED\_PICTURE**

|  |  |  |
| --- | --- | --- |
| Nume | Tip | Descriere |
| feed\_picture\_id | number(6) | Cheie primara |
| picture\_id | number(6) | Cheie externa, realizand legatura cu poza nou postata de un utilizator |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care vizualizeaza FEED-uri |

**FEED\_ALBUM**

|  |  |  |
| --- | --- | --- |
| Nume | Tip | Descriere |
| feed\_album\_id | number(6) | Cheie primara |
| album\_id | number(6) | Cheie externa, realizand legatura cu albumul nou postat de un utilizator |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care vizualizeaza FEED-uri |

**FEED\_STATUS**

|  |  |  |
| --- | --- | --- |
| Nume | Tip | Descriere |
| feed\_status\_id | number(6) | Cheie primara |
| status\_id | number(6) | Cheie externa, realizand legatura cu statusul nou postat de un utilizator |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care vizualizeaza FEED-uri |

**FEED\_EVENT**

|  |  |  |
| --- | --- | --- |
| Nume | Tip | Descriere |
| feed\_event\_id | number(6) | Cheie primara |
| event\_id | number(6) | Cheie externa, realizand legatura cu evenimentul nou creat de un utilizator |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care vizualizeaza FEED-uri |

**NOTIFICATION**

|  |  |  |
| --- | --- | --- |
| Nume | Tip | Descriere |
| notification\_id | number(6) | Cheie primara |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care posteaza comentarii |
| notification\_type\_id | number(6) | Cheie externa, realizand legatura cu tipul notificarii |
| notification\_type | nvarchar2(255) | Tipul notificarii |

**NOTIFICATION\_COMMENT**

|  |  |  |
| --- | --- | --- |
| Nume | Tip | Descriere |
| notification\_comment\_id | number(6) | Cheie primara |
| comment\_id | number(6) | Cheie externa, realizand legatura cu comentariul nou ce a fost adaugat la postarile unui utilizator. |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care posteaza comentarii |
| content | nvarchar2(255) | Continutul notificarii |

**NOTIFICATION\_LIKE**

|  |  |  |
| --- | --- | --- |
| Nume | Tip | Descriere |
| notification\_like\_id | number(6) | Cheie primara |
| like\_id | number(6) | Cheie externa, realizand legatura cu LIKE-ul dat de un utilizator la postarile altui utilizator. |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care da LIKE-uri |
| content | nvarchar2(255) | Continutul notificarii |

**NOTIFICATION\_FRIEND**

|  |  |  |
| --- | --- | --- |
| Nume | Tip | Descriere |
| notification\_friend\_id | number(6) | Cheie primara |
| friend\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care a trimis cererea de prietenie. |
| user\_id | number(6) | Cheie externa, realizand legatura cu utilizatorul care accepta cereri de prietenie |
| content | nvarchar2(255) | Continutul notificarii |

1. Scheme relationale

**USER** (#user\_id, language\_id, username, password, first\_name, last\_name, email, active, created\_at)

**PROFILE** (#profile\_id, user\_id, privacy\_id, city\_id, date\_of\_birth, about\_me, relationship, phone, interests, education, hobbies, fav\_movies, fav\_artists, fav\_books, address)

**CITY** (#city\_id, county\_id, privacy\_id, name)

**COUNTY** (#county\_id, country\_id, privacy\_id, name)

**COUNTRY** (#country\_id, privacy\_id, name)

**LANGUAGE** (#language\_id, name)

**FRIEND\_LIST** (#friend\_list\_id, user\_id, friend\_user\_id, state, created\_at)

**MESSAGE** (#message\_id, user\_id, friend\_user\_id, message\_content, created\_at, is\_read, is\_spam)

**STATUS** (#status\_id, user\_id, privacy\_id, status\_content, created\_at, updated\_at, numberOfLikes)

**PRIVACY** (#privacy\_id, privacy\_type)

**ALBUM** (#album\_id, user\_id, privacy\_id, name, description, created\_at, updated\_at, numberOfLikes)

**PICTURE** (#picture\_id, album\_id, privacy\_id, description, created\_at, updated\_at, is\_album\_cover, is\_profile\_picture, numberOfLikes)

**COMMENT\_STATUS** (#comment\_status\_id, user\_id, status\_id, comment\_content, created\_at, updated\_at, numberOfLikes)

**COMMENT\_ALBUM** (#comment\_album\_id, user\_id, album\_id, comment\_content, created\_at, updated\_at, numberOfLikes)

**COMMENT\_PICTURE** (#comment\_picture\_id, user\_id, picture\_id, comment\_content, created\_at, updated\_at, numberOfLikes)

**COMMENT** (#comment\_id, user\_id, comment\_type\_id)

**LIKE\_STATUS** (#like\_status\_id, user\_id, status\_id, created\_at)

**LIKE\_COMMENT** (#like\_comment\_id, user\_id, comment\_id, created\_at)

**LIKE\_ALBUM** (#like\_album\_id, user\_id, album\_id, created\_at)

**LIKE\_PICTURE** (#like\_picture\_id, user\_id, picture\_id, created\_at)

**LIKE** (#like\_id, user\_id, like\_type\_id)

**EVENT** (#event\_id, #user\_id, city\_id, name, description, address, date, created\_at, updated\_at)

**PARTICIPATE\_EVENT** (#participate\_event\_id, event\_id, user\_id)

**FEED\_STATUS** (#feed\_status\_id, status\_id, user\_id)

**FEED\_ALBUM** (#feed\_album\_id, album\_id, user\_id)

**FEED\_PICTURE** (#feed\_picture\_id, picture\_id, user\_id)

**FEED\_EVENT** (#feed\_event\_id, event\_id, user\_id)

**FEED** (#feed\_id, user\_id, feed\_type\_id)

**NOTIFICATION\_COMMENT** (#notification\_comment\_id, comment\_id, user\_id, content)

**NOTIFICATION\_LIKE** (#notification\_like\_id, like\_id, user\_id, content)

**NOTIFICATION\_FRIEND** (#notification\_friend\_id, friend\_id, user\_id, content)

**NOTIFICATION** (#notification\_id, user\_id, notification\_type\_id)

1. Normalizare

Pentru a exemplifica normalizarea in cadrul proiectului nostru, am ales cazul in care entitatea PROFILE contine toate campurile ce descriu adresa (city, county, country).

Consideram tabelul urmator:

|  |  |
| --- | --- |
| **profile\_id** | **address** |
| 11 | Strada 11, Bucuresti, judetul: Bucuresti, tara: Romania |
| 12 | Strada 12, Busteni, judetul: Prahova, tara: Romania |
| 13 | Strada 13, Sinaia, judetul: Prahova, tara: Romania |
| 14 | Strada 14, Bucuresti, judetul: Bucuresti, tara: Romania |
| 15 | Strada 15, Roman, judetul: Neamt, tara: Romania |
| 16 | Strada 16, Piatra-Neamt, judetul: Neamt, tara: Romania |

Aducem tabelul in forma normala 1 (FN1):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **profile\_id** | **address** | **city** | **county** | **country** |
| 11 | Strada 11 | Bucuresti | Bucuresti | Romania |
| 12 | Strada 12 | Busteni | Prahova | Romania |
| 13 | Strada 13 | Sinaia | Prahova | Romania |
| 14 | Strada 14 | Bucuresti | Bucuresti | Romania |
| 15 | Strada 15 | Roman | Neamt | Romania |
| 16 | Strada 16 | Piatra-Neamt | Neamt | Romania |

Observam ca e deja si in forma normala 2 (FN2), dar pentru utilizatori diferiti, avem campurile city, county, country ce se repeta. Aducem tabelul in forma FN3:

|  |  |  |
| --- | --- | --- |
| **profile\_id** | **address** | **city\_id** |
| 11 | Strada 11 | 1 |
| 12 | Strada 12 | 2 |
| 13 | Strada 13 | 3 |
| 14 | Strada 14 | 1 |
| 15 | Strada 15 | 4 |
| 16 | Strada 16 | 5 |

|  |  |  |
| --- | --- | --- |
| **city\_id** | **name** | **county\_id** |
| 1 | Bucuresti | 1 |
| 2 | Busteni | 2 |
| 3 | Sinaia | 2 |
| 4 | Roman | 3 |
| 5 | Piatra-Neamt | 3 |

|  |  |  |
| --- | --- | --- |
| **county\_id** | **name** | **country\_id** |
| 1 | Bucuresti | 1 |
| 2 | Prahova | 1 |
| 3 | Neamt | 1 |

|  |  |
| --- | --- |
| **country\_id** | **name** |
| 1 | Romania |

1. Denormalizare

Pentru a exemplifica denormalizarea in cadrul proiectului nostru, am ales cazul in care pentru un utilizator nu am mai retinut totalitatea comentariilor date. Cu alte cuvinte, avem urmatoarele tabele:

|  |  |  |
| --- | --- | --- |
| **comment\_picture\_id** | **user\_id** | **picture\_id** |
| 1 | 1 | 11 |
| 2 | 2 | 12 |

|  |  |  |
| --- | --- | --- |
| **comment\_album\_id** | **user\_id** | **album\_id** |
| 1 | 1 | 21 |
| 2 | 2 | 22 |

|  |  |  |
| --- | --- | --- |
| **comment\_status\_id** | **user\_id** | **status\_id** |
| 1 | 1 | 31 |
| 2 | 2 | 32 |

Ni se cere sa aflam pentru un utilizator dat totalitatea comentariilor postate de acestea. Avand modelul exemplificat mai sus prin cele 3 tabele, ar trebui sa facem JOIN pentru fiecare dintre cele 3 tabele cu entitatea USER si apoi sa colectam rezultatul.

Solutia aceasta e aplicabila doar pentru volum mic de date, dar avand in vedere profilul aplicatiei noastre – retea de socializare – volumul de date va creste exponential.

Asadar, alegem varianta pastrarii de informatii redundante intr-un tabel suplimentar:

|  |  |  |
| --- | --- | --- |
| **comment \_id** | **user\_id** | **comment\_type\_id** |
| 1 | 1 | 11 |
| 2 | 1 | 21 |
| 3 | 1 | 31 |
| 4 | 2 | 12 |
| 5 | 2 | 22 |
| 6 | 2 | 32 |

Astfel, pentru a vizualiza comentariile postate de un utilizator va fi suficienta interogarea tabelului COMMENT, fara a mai fi necesare JOIN-uri multiple.

1. Crearea bazei de date

CREATE TABLE USER(

user\_id number(6),

language\_id number(6),

username nvarchar2(255),

password nvarchar2(255),

first\_name nvarchar2(255),

last\_name nvarchar2(255),

email nvarchar2(255),

active nchar(1),

created\_at date,

CONSTRAINT user\_id\_pk PRIMARY KEY (user\_id,

CONSTRAINT lang\_fk FOREIGN KEY(language\_id) references LANGUAGE(language\_id)

);

CREATE TABLE PROFILE (

profile\_id number(6),

user\_id number(6),

privacy\_id number(6),

city\_id number(6),

date\_of\_birth date,

about\_me nvarchar2(255),

relationship nvarchar2(255),

phone nvarchar2(255),

interests nvarchar2(255),

education nvarchar2(255),

hobbies nvarchar2(255),

fav\_movies nvarchar2(255),

fav\_artists nvarchar2(255),

fav\_books nvarchar2(255),

address nvarchar2(255),

CONSTRAINT profile\_id\_pk PRIMARY KEY (profile\_id),

CONSTRAINT user\_profile\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id)

CONSTRAINT prof\_priv\_fk FOREIGN KEY (privacy\_id) REFERENCES PRIVACY(privacy\_id),

CONSTRAINT prof\_city\_fk FOREIGN KEY (city\_id) REFERENCES CITY(city\_id)

);

CREATE TABLE CITY (

city\_id number(6),

county\_id number(6),

privacy\_id number(6) FOREIGN KEY REFERENCES PRIVACY(privacy\_id),

name nvarchar2(255),

CONSTRAINT city\_id\_pk PRIMARY KEY (city\_id),

CONSTRAINT county\_city\_fk FOREIGN KEY (county\_id) REFERENCES COUNTY(county\_id)

);

CREATE TABLE COUNTY (

county\_id number(6),

country\_id number(6),

privacy\_id number(6) FOREIGN KEY REFERENCES PRIVACY(privacy\_id),

name nvarchar2(255),

CONSTRAINT county\_id\_pk PRIMARY KEY (county\_id),

CONSTRAINT county\_country\_fk FOREIGN KEY (country\_id) REFERENCES COUNTRY(country\_id)

);

CREATE TABLE COUNTRY (

country\_id number(6),

privacy\_id number(6),

name nvarchar2(255),

CONSTRAINT country\_id\_pk PRIMARY KEY (country\_id),

CONSTRAINT prof\_country\_fk FOREIGN KEY (privacy\_id) REFERENCES PRIVACY(privacy\_id),

);

CREATE TABLE LANGUAGE (

language\_id number(6),

name nvarchar2(255),

CONSTRAINT language\_id\_pk PRIMARY KEY (language\_id)

);

CREATE TABLE FRIEND\_LIST(

friend\_list\_id number(6) PRIMARY KEY,

user\_id number(6) FOREIGN KEY REFERENCES USER(user\_id),

friend\_user\_id number(6) FOREIGN KEY REFERENCES USER(user\_id),

state nchar(1),

created\_at date

);

CREATE TABLE MESSAGE(

message\_id number(6),

user\_id number(6),

friend\_user\_id number(6),

message\_content nvarchar2(255),

created\_at date,

is\_read nchar(1),

is\_spam nchar(1),

CONSTRAINT message\_id\_pk PRIMARY KEY (message\_id),

CONSTRAINT mes\_user\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id),

CONSTRAINT mes1\_user\_FK FOREIGN KEY (friend\_user\_id) REFERENCES USER(user\_id)

);

CREATE TABLE STATUS (

status\_id number(6),

user\_id number(6),

privacy\_id number(6) FOREIGN KEY REFERENCES PRIVACY(privacy\_id),

status\_content NVARCHAR2(255),

created\_at date,

updated\_at date,

numberOfLikes int,

CONSTRAINT status\_pk PRIMARY KEY (status\_id),

CONSTRAINT status\_user\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id)

);

CREATE TABLE PRIVACY(

privacy\_id number(6),

privacy\_type nvarchar2(255),

CONSTRAINT priv\_pk PRIMARY KEY (privacy\_id)

);

CREATE TABLE PICTURE(

picture\_id number(6),

album\_id number(6),

privacy\_id number(6) FOREIGN KEY REFERENCES PRIVACY(privacy\_id),

description nvarchar2(255),

created\_at date,

updated\_at date,

is\_album\_cover nchar(1),

is\_profile\_picture nchar(1),

numberOfLikes int,

CONSTRAINT pic\_pk PRIMARY KEY (picture\_id),

CONSTRAINT pic\_album\_fk FOREIGN KEY (album\_id) REFERENCES ALBUM(album\_id)

);

CREATE TABLE ALBUM(

album\_id number(6),

user\_id number(6),

privacy\_id number(6) FOREIGN KEY REFERENCES PRIVACY(privacy\_id),

name nvarchar2(255),

description nvarchar2(255),

created\_at date,

updated\_at date,

numberOfLikes int,

CONSTRAINT album\_pk PRIMARY KEY (album\_id),

CONSTRAINT album\_user\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id)

);

CREATE TABLE COMMENT(

comment\_id number(6),

user\_id number(6) FOREIGN KEY REFERENCES USER(user\_id),

comment\_type nvarchar2(32), -- Type of the comment to know in what table need to look

comment\_type\_id number(6),

CONSTRAINT comm\_pk PRIMARY KEY (comment\_id)

);

CREATE TABLE COMMENT\_PICTURE(

comment\_picture\_id number(6),

user\_id number(6),

picture\_id number(6),

comment\_content nvarchar2(255),

created\_at date,

updated\_at date,

numberOfLikes int,

CONSTRAINT cp\_pk PRIMARY KEY (comment\_picture\_id),

CONSTRAINT cP\_comm\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id),

constraint cp\_pic\_fk FOREIGN KEY (picture\_id) REFERENCES PICTURE(picture\_id)

);

CREATE TABLE COMMENT\_ALBUM(

comment\_album\_id number(6),

user\_id number(6),

album\_id number(6),

comment\_content nvarchar2(255),

created\_at date,

updated\_at date,

numberOfLikes int,

CONSTRAINT ca\_pk PRIMARY KEY (comment\_album\_id),

CONSTRAINT ca\_comUSR\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id),

CONSTRAINT ca\_album\_fk FOREIGN KEY (album\_id) REFERENCES ALBUM(album\_id)

);

CREATE TABLE COMMENT\_STATUS(

comment\_status\_id number(6),

user\_id number(6),

status\_id number(6),

comment\_content nvarchar2(255),

created\_at date,

updated\_at date,

numberOfLikes int,

CONSTRAINT cs\_pk PRIMARY KEY (comment\_status\_id),

CONSTRAINT cs\_comUSR\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id),

CONSTRAINT status\_cs\_fk FOREIGN KEY (status\_id) REFERENCES STATUS(status\_id)

);

CREATE TABLE LIKE(

like\_id number(6) CONSTRAINT l\_nn NOT NULL,

user\_id number(6),

like\_type\_id number(6),

like\_type nvarchar2(32),

CONSTRAINT l\_pk PRIMARY KEY (like\_id),

CONSTRAINT l\_fr\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id)

);

CREATE TABLE LIKE\_PICTURE(

like\_picture\_id number(6),

user\_id number(6),

picture\_id number(6),

created\_at date,

CONSTRAINT lp\_pk PRIMARY KEY (like\_picture\_id),

CONSTRAINT l\_lp\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id),

CONSTRAINT lp\_p\_fk FOREIGN KEY (picture\_id) REFERENCES PICTURE(picture\_id)

);

CREATE TABLE LIKE\_ALBUM(

like\_album\_id number(6),

user\_id number(6),

album\_id number(6),

created\_at int,

CONSTRAINT la\_pk PRIMARY KEY (like\_album\_id),

CONSTRAINT l\_la\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id),

CONSTRAINT la\_a\_fk FOREIGN KEY (album\_id) REFERENCES ALBUM(album\_id)

);

CREATE TABLE LIKE\_STATUS(

like\_status\_id number(6),

user\_id number(6),

status\_id number(6),

created\_at int,

CONSTRAINT ls\_pk PRIMARY KEY (like\_status\_id),

CONSTRAINT ls\_l\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id),

CONSTRAINT st\_l\_fk FOREIGN KEY (status\_id) REFERENCES STATUS(status\_id)

);

CREATE TABLE LIKE\_COMMENT(

like\_comment\_id number(6),

user\_id number(6),

comment\_id number(6),

created\_at int,

CONSTRAINT lk\_pk PRIMARY KEY (like\_comment\_id),

CONSTRAINT l\_lk\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id),

CONSTRAINT c\_lk\_fk FOREIGN KEY (comment\_id) REFERENCES COMMENT(comment\_id)

);

CREATE TABLE EVENT(

event\_id number(6),

user\_id number(6) FOREIGN KEY REFERENCES USER(user\_id),

city\_id number(6) FOREIGN KEY REFERENCES CITY(city\_id),

name nvarchar2(255),

description nvarchar2(255),

address number(6),

date date,

created\_at date,

updated\_at date,

CONSTRAINT e\_pk PRIMARY KEY(event\_id),

);

CREATE TABLE PARTICIPATE\_EVENT(

participate\_event\_id number(6),

event\_id number(6),

user\_id number(6),

CONSTRAINT ev\_p\_pk PRIMARY KEY (p\_event\_id),

CONSTRAINT ev\_p\_fk FOREIGN KEY (event\_id) REFERENCES EVENT(event\_id),

CONSTRAINT us\_p\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id)

);

CREATE TABLE FEED(

feed\_id number(6),

user\_id number(6),

feed\_type\_id number(6),

CONSTRAINT feed\_pk PRIMARY KEY (feed\_id),

CONSTRAINT feed\_u\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id)

);

CREATE TABLE FEED\_PICTURE(

feed\_picture\_id number(6),

picture\_id number(6),

user\_id number(6),

CONSTRAINT feed\_p\_pk PRIMARY KEY (feed\_picture\_id),

CONSTRAINT fp\_pic\_fk FOREIGN KEY (picture\_id) REFERENCES PICTURE(picture\_id),

CONSTRAINT fp\_feed\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id)

);

CREATE TABLE FEED\_ALBUM(

feed\_album\_id number(6),

album\_id number(6),

user\_id number(6),

CONSTRAINT feed\_a\_pk PRIMARY KEY (feed\_album\_id),

CONSTRAINT fa\_al\_fk FOREIGN KEY (album\_id) REFERENCES ALBUM(album\_id),

CONSTRAINT fp\_feed\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id)

);

CREATE TABLE FEED\_STATUS(

feed\_status\_id number(6),

status\_id number(6),

user\_id number(6),

CONSTRAINT feed\_s\_pk PRIMARY KEY (feed\_status\_id),

CONSTRAINT fs\_usr\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id),

CONSTRAINT fs\_st\_fk FOREIGN KEY (status\_id) REFERENCES STATUS(status\_id),

);

CREATE TABLE FEED\_EVENT(

feed\_event\_id number(6),

event\_id number(6),

user\_id number(6),

CONSTRAINT feed\_e\_pk PRIMARY KEY (feed\_event\_id),

CONSTRAINT fe\_usr\_fk FOREIGN KEY (user\_id) REFERENCES USER(user\_id),

CONSTRAINT fe\_ev\_fk FOREIGN KEY (event\_id) REFERENCES EVENT(event\_id)

);

CREATE TABLE NOTIFICATION(

notification\_id number(6) PRIMARY KEY,

user\_id number(6) FOREIGN KEY REFERENCES USER(user\_id),

notification\_type\_id number(6),

notification\_type nvarchar2(32)

)

CREATE TABLE NOTIFICATION\_COMMENT(

notification\_comment\_id number(6),

comment\_id number(6),

user\_id number(6) FOREIGN KEY REFERENCES USER(user\_id),

content nvarchar2(255),

CONSTRAINT not\_comm\_pk PRIMARY KEY (notification\_comment\_id),

CONSTRAINT com\_not\_fk FOREIGN KEY (comment\_id) REFERENCES COMMENT(comment\_id)

);

CREATE TABLE NOTIFICATION\_LIKE(

notification\_like\_id number(6),

like\_id number(6),

user\_id number(6) FOREIGN KEY REFERENCES USER(user\_id),

content nvarchar2(255),

CONSTRAINT not\_lk\_pk PRIMARY KEY (notification\_like\_id),

CONSTRAINT lik\_not\_fk FOREIGN KEY (like\_id) REFERENCES LIKE(like\_id)

);

CREATE TABLE NOTIFICATION\_FRIEND(

notification\_firend\_id number(6),

friend\_id number(6) FOREIGN KEY REFERENCES USER(user\_id),

user\_id number(6) FOREIGN KEY REFERENCES USER(user\_id),

content nvarchar2(255),

CONSTRAINT not\_fr\_pk PRIMARY KEY (notification\_friend\_id)

);

1. Ce se doreste de la model

Modelul nostru isi propune sa implementeze functionalitatile de baza ale unei retele de socializare. Dintre acestea amintim :

* Posibilitatea crearii unui cont de utilizator si completarea unui profil
* Posibilitatea editarii tuturor informatiilor prezente in cadrul profilului
* Vizualizarea profilelor altor utilizatori
* Postarea, editarea si stergerea de statusuri, poze, albume
* Publicarea, editarea, stergerea comentariilor si/ sau aprecierea anumitor continuturi apartinand altor utilizatori din lista de prieteni
* Trimiterea cererilor de prietenie catre alti utilizatori, urmata sau nu de aprobarea acestora
* Comunicarea, prin intermediul mesajelor, cu utilizatorii din lista de prieteni
* Crearea si participarea la evenimente
* Notificarea utilizatorului asupra interactiunii altor utilizatori cu continutul postat de el
* Accesarea ultimelor noutati in ceea ce ii priveste pe utilizatorii din lista de prieteni

1. Diagrama conceptuala a modelului

